

(19)



Europäisches Patentamt

European Patent Office

Office européen des brevets



(11)

EP 0 725 511 A2

(12)

## EUROPEAN PATENT APPLICATION

(43) Date of publication:  
07.08.1996 Bulletin 1996/32

(51) Int Cl.<sup>6</sup>: H04L 9/06

(21) Application number: 96300526.9

(22) Date of filing: 25.01.1996

(84) Designated Contracting States:  
DE FR GB

• Rogaway, Phillip W.  
Davis, California 95616 (US)

(30) Priority: 06.02.1995 US 384152

(74) Representative: Jennings, Michael John  
IBM United Kingdom Limited,  
Intellectual Property Department,  
Hursley Park  
Winchester, Hampshire SO21 2JN (GB)

(71) Applicant: International Business Machines  
Corporation  
Armonk, N.Y. 10504 (US)

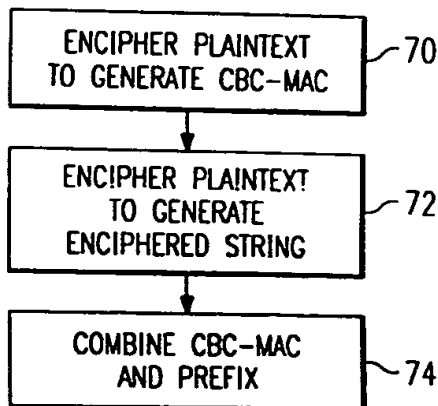
(72) Inventors:  
• Bellare, Mihir  
Hawthorne, New York 10532 (US)

(54) **Method for data encryption/decryption using cipher block chaining (CBC) and message authentication codes (MAC)**

(57) A method for encrypting a plaintext string into ciphertext begins by cipher block chaining (CBC) (70) the plaintext using a first key and a null initialization vector to generate a CBC message authentication code (MAC) whose length is equal to the block length. The plaintext string is then cipher block chained (72) again, now using a second key and the CBC-MAC as the ini-

tialization vector, to generate an enciphered string. The CBC-MAC and a prefix of the enciphered string comprising all of the enciphered string except the last block are then combined (74) to create the ciphertext. The described mode of operation is length-preserving, yet has the property that related plaintexts give rise to unrelated ciphertexts.

FIG. 3



EP 0 725 511 A2

**Description****TECHNICAL FIELD**

5 The present invention relates generally to secure communications and more particularly to computer-implemented methods to encrypt plaintext into ciphertext.

**BACKGROUND OF THE INVENTION**

10 It has been generally accepted that encryption schemes exhibiting certain properties are often desirable. The first of these properties is that the encrypting and decrypting operations are deterministic, as opposed to probabilistic, because in many environments there is no available or trustworthy source of randomness. It is also desirable that the scheme be history-free (not stateful) so that parties need not store a message counter or other information that must be updated after each encryption or decryption. The scheme should also be "secure" in that it effectively hides  
15 all information about the plaintext. Lastly, it is desired that the scheme be length-preserving, i.e. the length of the ciphertext should equal the length of the plaintext.

Block ciphers are well-known cryptographic tools that are often used to implement general encryption schemes. A block cipher is a symmetric key cryptosystem that transforms message (plaintext) blocks of fixed length (of " $l$ " bits) into ciphertext blocks of the same length under the control of a key (of " $k$ " bits). A widely used block cipher is provided  
20 by the U.S. Standard DES algorithm, which has  $l = 64$  and  $k = 56$ , and is described in NBS FIPS Pub 46, titled "Data Encryption Standard", National Bureau of Standards, U.S. Department of Commerce, January 1977. Block ciphers like DES provide a way to encrypt a single block (e.g., 64-bits) of text. But to encrypt longer messages, the block cipher must be used in some "mode of operation." Many such modes of operation have been described in the prior art, with the most widely used one being Cipher Block Chaining (CBC). CBC is described in NBS FIPS Pub 81, titled "DES  
25 Modes of Operation", National Bureau of Standards, U.S. Department of Commerce, December 1980. CBC and other known modes, however, are either length-increasing or suffer from the weakness that distinct related plaintexts give rise to related ciphertexts. Many application domains that cannot tolerate the former have their security effectively compromised by the latter.

Cipher Block Chaining (CBC) requires the use of a secret key as well an "initialization vector" (IV). With an  $l$ -bit  
30 IV (the value of which is sent with the message or is otherwise known by both communicating parties), a string  $x = x_1 \dots x_n$  (consisting of  $n$  blocks, each of  $l$  bits) is then encrypted as  $E_{a,IV}(x) = Y_1 \dots Y_n$ , where  $y_0 = IV$  and  $y_i = f_a(x_i \oplus y_{i-1})$ . In a CBC scheme, the first block of the ciphertext depends on the first block of the plaintext, the second block of the ciphertext depends on the first two blocks of the plaintext, and so on, with the last block of the ciphertext depending on all of the blocks of the plaintext. Such encryption, however, has a well-known drawback in that it is not secure  
35 enough when IV is fixed.

In particular, the CBC method often "leaks" information about plaintexts that are being encrypted. For example, if an adversary sees  $E_{a,IV}(X)$  and  $E_{a,IV}(X')$ , and notices that they agree in the first  $j$  blocks, then the adversary can infer that  $X$  and  $X'$  also agree in the first  $j$  blocks. Such deficiencies are quite problematic. Thus, suppose that a file consisting of a sequence of 1 KByte employee records is noticed to have just changed from the 7th record on. Perhaps it is  
40 known *a priori* that the reason for this change was the update of an employee record due to someone having been demoted. If the underlying encryption method is  $E_{a,IV}$ , and the employee records are in alphabetical order by employee name, then one can infer that the affected employee is the 7th one in alphabetical order.

The above-described characteristic of CBC encryption to "leak" information about plaintexts could be addressed by choosing the initialization vector IV at random and then sending it along with the message. However, when this is  
45 done the scheme is no longer length-preserving. Alternatively, the encryption of a message could be made history-dependent (e.g., by using IV as a function of a message counter and not sending IV with the message), but this approach is also unsatisfactory because it is intolerant of non-receipt of messages by the intended recipient.

Thus, prior art encryption techniques that use block ciphers are undesirable in that they are length-increasing, intolerant of messages being dropped, or leak information about related plaintexts. There remains a need to provide  
50 a secure, length-preserving encryption scheme using block ciphers that overcomes these and other problems in the art.

**SUMMARY OF THE INVENTION**

According to a first aspect of the invention there is provided a method of encrypting a plaintext string that is  
55 deterministic and history-free.

A preferred method uses first and second secret keys, to encrypt a plaintext string to a ciphertext string. The method begins by cipher block chaining the plaintext string using the first key and a fixed initialization vector to generate a CBC message authentication code (CBC-MAC) of length equal to the block length. Thereafter, the method continues

by cipher block chaining the plaintext string using the second key, and using the aforementioned CBC message authentication code as the initialization vector, to thereby generate an enciphered string. The CBC message authentication code and a prefix of the enciphered string are then combined (typically by concatenation) to form the ciphertext string. Preferably, the technique is length-preserving; the prefix includes all but the final block so that the length of the ciphertext is equal to the length of the plaintext.

The invention thus preferably provides a block cipher mode of operation for encryption wherein the length of the ciphertext is the same as the length of the plaintext being encrypted. Preferably, such a length-preserving encryption scheme does not leak information about plaintexts that are being encrypted.

It is preferred that message encryption schemes according to the invention are history-free, so that parties do not store a message counter or other information that must be updated after each encryption or decryption.

It is further preferred that length-preserving encryption schemes according to the invention apply Cipher Block Chaining (CBC) in a way that overcomes the known security and information leakage problems associated with CBC encryption. This technique is highly advantageous in that modification of a ciphertext message to a ciphertext message not yet seen produces the encryption of an underlying message unrelated to those then seen.

The invention preferably provides methods of encrypting plaintext message strings that have lengths that are multiples of or fractions of a block length.

Thus according to the preferred method the plaintext string is processed using CBC twice, first to generate the CBC-MAC, and then to generate a portion of the ciphertext itself. In the first pass, the initialization vector used in the CBC is the null vector (meaning a string of 0-bits having length equal to the block length). In the second pass, the initialization vector is the CBC-MAC generated in the first pass. The keys are distinct for the two passes. The method is useful for generating ciphertext when the plaintext string has a length that is a multiple of a length of a block. A variant of the scheme can be used when the plaintext string has a length that is a fraction of the block length.

To decrypt the ciphertext, the enciphered string portion thereof is cipher block chained using the second key and the CBC-MAC as the initialization vector to generate a deciphered string. The deciphered string is then cipher block chained using the first key and a null IV to generate a string having a last block. The plaintext is then taken as the combination (e.g., by concatenation) of the deciphered string and a predetermined function (e.g., an XOR) of the last block, and the inverse of the block cipher under the first key at the CBC-MAC.

The invention may be implemented in a programmed computer or in dedicated hardware or software. In one embodiment, the various methods of the invention may be implemented on a program storage device (e.g., a floppy diskette) that is readable by a processor and that tangibly embodies a program of instructions executable by the processor to perform the various process steps of each method.

According to another aspect of the invention, there is provided a computer-implemented method, using first and second keys, to encrypt a plaintext string  $x$  to a ciphertext string  $y$ , comprising the steps of:

using the string  $x$  and the first key  $a_0$  to compute a message authentication code  $t$ ;  
using the string  $x$ , the second key  $a_1$ , and the message authentication code  $t$  to produce an enciphered string  $y'$  that depends substantively on the message authentication code; and  
taking the ciphertext  $y$  to comprise the message authentication code  $t$  together with a predetermined piece of the enciphered string  $y'$ .

Preferably, the predetermined piece is shorter than  $y'$  yet the plaintext string is still uniquely recoverable given the ciphertext. The message authentication code is preferably computed by cipher block chaining a block cipher (e.g. the DES block cipher).

A third aspect of the invention provides a computer apparatus, comprising:

a storage device;  
program means supported in the storage device for encrypting a plaintext string  $x$  to a ciphertext string  $y$ , the program means comprising:

means for using the string  $x$  and a first key  $a_0$  to compute a message authentication code  $t$ ;  
means for using the string  $x$ , a second key  $a_1$ , and the message authentication code  $t$  to produce an enciphered string  $y'$ ; and  
means for taking the ciphertext  $y$  to comprise the message authentication code  $t$  together with a predetermined piece of the enciphered string  $y'$ , where said predetermined piece is shorter than  $y'$ .

## BRIEF DESCRIPTION OF THE DRAWINGS

The invention will now be described in more detail, by way of example, with reference to the accompanying drawings

in which:

FIGURE 1 illustrates a computer comprising a system unit, a keyboard, a mouse and a display, for use in implementing the encryption and decryption methods of the present invention;

FIGURE 2 is an architectural block diagram of the computer illustrated in FIGURE 1;

FIGURE 3 illustrates a simplified flow diagram of a method of the invention for encrypting a plaintext into ciphertext;

FIGURE 4 illustrates a simplified flow diagram illustrating how the ciphertext (generated in FIGURE 3) is converted back to plaintext;

FIGURE 5A illustrates step 70 of FIGURE 3;

FIGURE 5B illustrates step 72 of FIGURE 3;

FIGURE 5C illustrates step 76 of FIGURE 4;

FIGURE 5D illustrates step 78 of FIGURE 4; and

FIGURE 5E illustrates step 80 of FIGURE 4.

## DETAILED DESCRIPTION

By way of brief background, a computer for use in supporting the invention is shown in FIGURE 1. The computer 20 comprises a system unit 21, a keyboard 22, a mouse 23 and a display 24. The screen 26 of display device 24 is used to present a graphical user interface (GUI). The graphical user interface supported by the operating system allows the user to use a point and shoot method of input, i.e., by moving the mouse pointer 25 to an icon representing a data object at a particular location on the screen 26 and pressing on the mouse buttons to perform a user command or selection.

FIGURE 2 shows a block diagram of the components of the personal computer shown in FIGURE 1. The system unit 21 includes a system bus or plurality of system buses 31 to which various components are coupled and by which communication between the various components is accomplished. The microprocessor 32 is connected to the system bus 31 and is supported by read only memory (ROM) 33 and random access memory (RAM) 34 also connected to system bus 31. A microprocessor in the IBM PS/2 series of computers is one of the Intel family of microprocessors including the 386 or 486 microprocessors. Other microprocessors, including but not limited to, Motorola's family of microprocessors such as the 68000, 68020 or the 68030 microprocessors and various RISC microprocessors such as the PowerPC microprocessor manufactured by IBM, and others made by Hewlett Packard, Sun, Intel, Motorola and others may be used in the specific computer.

The ROM 33 contains among other code the Basic Input-Output system (BIOS) which controls basic hardware operations such as the interaction and the disk drives and the keyboard. The RAM 34 is the main memory into which the operating system and application programs are loaded. The memory management chip 35 is connected to the system bus 31 and controls direct memory access operations including, passing data between the RAM 34 and hard disk drive 36 and floppy disk drive 37. The CD ROM 42, also coupled to the system bus 31, is used to store a large amount of data, e.g., a multimedia program or large database.

Also connected to this system bus 31 are various I/O controllers: the keyboard controller 38, the mouse controller 39, the video controller 40, and the audio controller 41. The keyboard controller 38 provides the hardware interface for the keyboard 22, the mouse controller 39 provides the hardware interface for the mouse 23, the video controller 40 is the hardware interface for the display 24, and the audio controller 41 is the hardware interface for the speakers 25a and 25b. An I/O controller 50 such as a Token Ring Adapter enables communication over the local area network 56 to other similarly configured data processing systems.

One of the preferred implementations of the present invention is as a set of instructions in a code module resident in the random access memory 34. Until required by the computer system, the set of instructions may be stored in another computer memory, for example, in the hard disk drive 36, or in a removable memory such as an optical disk for eventual use in the CD ROM 42 or in a floppy disk for eventual use in the floppy disk drive 37. In addition, although the various methods described are conveniently implemented in a general purpose computer selectively activated or reconfigured by software, one of ordinary skill in the art would also recognize that such methods may be carried out in hardware, in firmware, or in more specialized apparatus constructed to perform the required method steps.

As used herein, the inventive method is designed to be implemented on a computer such as shown in FIGURE 1 although it should be appreciated that the word "computer" is to be afforded its broadest scope and meaning to include any type of device or part thereof that provides a computing functionality regardless of the particular application.

Turning now to FIGURE 3, the preferred method for encrypting a plaintext string into ciphertext is illustrated by means of a flow diagram. It is assumed that the encrypting party and the decrypting party share a pair of secret keys (i.e. a first and a second key). At step 70, the plaintext string is cipher block chained using the first (secret) key and a null initialization vector (IV) to generate a CBC message authentication code (MAC) that is the (entire) last block of ciphertext. At step 72, the plaintext string is again cipher block chained, now using the second (secret) key and the

CBC-MAC (generated in step 70) as the initialization vector, to thereby generate an enciphered string. At step 74, the CBC-MAC (generated in step 70) and a portion of the enciphered string (generated in step 72) are then combined to create the ciphertext. The portion of the enciphered string is also referred to as a "prefix". This combination is further a function of the first key.

Decryption of the ciphertext (generated by the routine of FIGURE 3) is illustrated in FIGURE 4. At step 76, the enciphered string (generated in step 72) is decrypted by cipher block chaining using the second secret key and the CBC-MAC (generated in step 70) as the initialization vector. Step 76 generates a deciphered string. At step 78, the deciphered string is then cipher block chained using the first key and a null IV to generate another string having a last block. At step 80, a predetermined function of this last block and the inverse of the block cipher at the CBC-MAC (generated in step 70) under the first key is then calculated. The plaintext is then formed at step 82 as the combination (i.e., the concatenation) of the deciphered string and the result of the predetermined function.

The operations in each of the steps 70 and 72 of the encryption routine are illustrated in FIGURES 5A and 5B, respectively. The routine uses an  $t$ -bit block cipher  $f$  (like DES) with key length  $k$ . We write  $f_a(x)$  for the  $t$ -bit string which is the block cipher's value applied to the  $t$ -bit  $x$  using the  $k$ -bit key  $a$ . Further, as noted above, it is assumed at the outset that the first and secret keys  $a_0$  and  $a_1$  are available to the routine and that  $la_0l = la_1l = k$ . The keys can be derived from some underlying  $k$ -bit key  $K$  using standard key separation techniques. For example,  $a_0$  could be the first  $k$  bits of  $f_K(0)$  and  $a_1$  could be the first  $k$  bits of  $f_K(1)$ . In FIGURE 5A, the plaintext string consists of the message string  $x$ , which for illustrative purposes is assumed to be comprised of ten (10) blocks of sixty-four (64) bits each, or 640 bits total. The message string is thus  $x = x_1 x_2 \dots x_{10}$ . This string is applied to the cipher block chaining encryption routine 82, which also receives the first key  $a_0$  and a null initialization vector (i.e.,  $IV = 0$ ). The result of the cipher block chaining routine 82 is an output string  $y = y_1 y_2 \dots y_{10}$ . The last block  $y_{10}$  is the 64-bit cipher block chaining message authentication code or "CBC-MAC". This completes the first pass of the routine.

The second pass is shown in FIGURE 5B wherein the message string (i.e., the plaintext) is again supplied to the cipher block chaining encryption routine 82. However, in this pass, the key used by the routine is the second (secret) key  $a_1$ , and the initialization vector is the CBC-MAC (i.e.,  $y_{10}$ ) generated in the first pass illustrated in FIGURE 5A. The resulting enciphered string is called  $y' = y'_1 y'_2 \dots y'_{10}$ . This processing completes the second pass. Note that although the block cipher  $f$  shown in FIGURES 5A and 5B is shown to be the same, this is not required. The ciphertext is then taken to be the combination (e.g., by concatenation) of the CBC-MAC and a portion of the enciphered string, namely:

$$\text{ciphertext} = y_{10} | y'_1 y'_2 \dots y'_9$$

The routine is length preserving since the length of the ciphertext is the same as the length of the plaintext string.

To decrypt the 10-block string  $y$ , we first consider it to be a sequence of blocks:

$$y_{10} | y'_1 y'_2 \dots y'_9$$

The operations in each of the steps 76, 78 and 80 of the decryption routine are then as illustrated in FIGURES 5C, 5D and 5E, respectively. As shown in FIGURE 5C, step 76 involves CBC decryption 84 of the enciphered string  $y'_1 y'_2 \dots y'_9$  (generated in step 72) with the second key  $a_1$  and the CBC-MAC (i.e.,  $y_{10}$ ) as the IV. The resulting deciphered string is  $x_1 x_2 \dots x_9$ , which represents almost all of the original plaintext. To recover  $x_{10}$ , the decryption routine first carries out the operation shown in FIGURE 5D, wherein the deciphered string  $x_1 x_2 \dots x_9$  is cipher block encrypted (by CBC 84) using the first key  $a_0$  and a null IV to generate a string  $y_1 y_2 \dots y_9$  having a last block  $y_9$ . As seen in FIGURE 5E, a predetermined function 86 (e.g., an XOR) of  $y_9$  and an inverse function of the block cipher  $f$  under the first key  $a_0$  at the point  $y_{10}$  is then calculated to generate  $x_{10}$ . The plaintext is then seen as the following:

$$\text{plaintext} = x_1 x_2 \dots x_9 | x_{10}$$

The preferred implementation illustrated above utilizes cipher block chaining as the mode of operation for the block cipher in steps 72 and 76. The invention, however, is not so limited, as other modes of operation may also be used for these steps. Moreover, although cipher block chaining is preferably used in the first pass (step 70) over the plaintext to create the message authentication code, it should be appreciated that other known techniques for producing MAC's (or other block cipher chaining modes) could be substituted in this step instead of CBC. (All that is necessary is that, given the  $t$ -bit MAC of  $m$  and all but particular  $t$ -bits of  $m$ , those missing  $t$ -bits can be efficiently and uniquely reconstructed). Thus, according to the invention it is envisioned that the first pass that processes the plaintext string involves a known technique that uses the first key  $a_0$  for computing a message authentication code or tag. As discussed above, the second pass then involves using this MAC as an IV along with a second key  $a_1$  to encrypt the message into an enciphered string. This second pass can be performed using CBC, but this is not required. The MAC and a portion of the enciphered string is then taken as the ciphertext.

Thus, in accordance with the more general aspects of the invention, encryption involves using the plaintext string and a first key to compute a message authentication code. The routine continues by using the message, a second key, and the message authentication code to produce an enciphered string that depends substantively on the message

authentication code. As used herein, such "substantive" dependence means that all bits of the enciphered string may vary as the MAC takes on different values. The ciphertext of the plaintext is then taken to comprise the message authentication code together with some piece of the enciphered string. To reverse the process, the decryption routine involves using the enciphered string portion of the ciphertext, the second key and the MAC to generate a deciphered string. Decryption continues by using the deciphered string and the first key to produce a string having a last block. A predetermined function of the last block and the inverse of the block cipher under the first key at the MAC is then computed. The plaintext is then taken as the deciphered string and the result of the predetermined function.

A more detailed implementation of the invention is now set forth. This implementation processes message strings whether or not the length of the particular string being processed is equal to or a fraction of a desired block length. The method begins by selecting a  $l$ -bit block cipher  $f$  with key length  $k$ . For example,  $l$  is 64 when  $f$  is the DES algorithm. Of course, other block ciphers (e.g., IDEA or SKIPJACK) besides DES may be used as well. Let the encryption key be  $a = (a_0, a_1)$ , with  $|a_0| = |a_1| = k$ , and let  $\lambda$  denote the empty string (with  $0^\circ = 1$ ). Secret keys  $a_0$  and  $a_1$  should be unrelated to each other (at least with respect to practical computation). Let  $\langle m \rangle$  denote the encoding of the number  $m < 2^l$  into an  $l$ -bit block. For a string  $m = m_1 \dots m_s$  consisting of  $s$  blocks, each of  $l$ -bits, the  $(l$ -bit) CBC-MAC of  $m$  under  $a_0$  is then defined by:

$$f^{(s)}(m) = f_{a_0}(f_{a_0}(\dots(f_{a_0}(m_1) \oplus m_2) \oplus \dots \oplus m_{s-1}) \oplus m_s) \oplus a_0$$

Now suppose  $x$  is the message we want to encrypt and  $1 \leq |x| \leq 2^l$ . Let  $x = x_1 \dots x_{n-1} x_n$  be the message to be encrypted, with  $|x_1| = \dots = |x_{n-1}| = 1$  and  $|x_n| \leq 1$ . Note the assumption  $|x_n| \geq 1$  implies that there is at least one "full" block to encrypt. The following method is not to be applied to messages of length less than 1. Let

$$x^* = \langle |x| \rangle x_1 \dots x_{n-3} x_{n-2} x_n 0^{1-x_n} 1^{x_n}$$

The above step first pads the message string with trailing zeros to insure that the overall length of the string being encrypted is a multiple of the block length, and then it swaps the (previously short) last block of  $x$  with the second-to-last (full) block of  $x$ . (This step is not required if the length of the message is a multiple of the block length). Now let

$$x^* = x_1 \dots x_{n-2} x_n$$

Note that  $|x^*| = |x| = 1$ .

The encryption scheme  $E_a(\cdot)$  is as follows:

- o Step 1. Let  $t = f_{a_0}^{(n+1)}(x)$  be the  $l$ -bit CBC-MAC of  $x$  under  $a_0$ .
- o Step 2. Encipher  $x^*$  as follows. Let  $y_0 = t$  (i.e., the initialization vector). Then for  $i = 1, \dots, n-2$  let  $y_i = f_{a_1}(x_i \oplus y_{i-1})$ . Finally, if  $|x_n| = 1$  let  $y_n = f_{a_1}(x_n \oplus y_{n-2})$ ; else (i.e.  $1 \leq |x_n| < 1$ ) let  $y_n$  be the XOR of  $x_n$  with the first  $|x_n|$  bits of  $f_{a_1}(y_{n-2})$ . This encryption method is an extension of the CBC mode of operation to allow for variable-length blocks. (When the block cipher is DES, this method has been called the IBM CUSP/3848 mechanism).
- o Step 3. Define  $E_a(x) = t y_1 \dots y_{n-2} y_n$ . That is, the encryption of  $x$  is  $t$  together with the enciphered text from Step 2.

Decryption is done as follows, with  $y = t y_1 \dots y_{n-2} y_n$  the received ciphertext:

- o Step 1. Recover  $x^*$  by deciphering under key  $a_1$  the ciphertext  $y_1 \dots y_{n-1} y_n$ . That is, let  $y_0 = t$  and for  $i = 1, \dots, n-2$  let  $x_i = f^{-1}(y_i) \oplus y_{i-1}$ . Then, if  $|y_n| = 1$ , then let  $x_n = a_1 f^{-1}(y_n) \oplus y_{n-2}$ ; else (i.e.  $1 \leq |y_n| < 1$ ) let  $x_n$  be the  $a_1$  XOR of  $y_n$  with the first  $|y_n|$  bits of  $f_{a_1}(y_{n-2})$ .
- o Step 2. To recover  $x_{n-1}$ , let  $t = f_{a_0}^{(n)}(|y| x_1 \dots x_{n-2} x_n 0^{1-x_n} 1^{x_n})$  and let  $x_{n-1} = f_{a_0}^{-1}(t) \oplus t$ .

The recovered plaintext is  $x_1 \dots x_n$ .

The present invention provides significant advantages in that the encryption is length-preserving, non-stateful, deterministic and secure. Related encryption methods can be designed in order to achieve not only these initial set of requirements but further to insure that the methods are fully parallelizable, either in hardware or in software. For example, the MAC  $t$  can be computed by a "tree MAC" scheme as described in U.S. Patent No. 4,933,969 to Marshall et al, incorporated herein by reference. Then, the encipherment may involve simply XORing the message  $x$  with the length -  $|x|$  prefix of  $f_{a_1}(t) f_{a_1}((t+1) \bmod 2^l) f_{a_1}((t+2) \bmod 2^l) \dots$ . Under such an embodiment, doubling the number of proces-

sors effectively doubles the rate at which the enciphered text can be computed.

The particular applications of the methods detailed herein are quite varied. For example, the techniques are useful for encrypting a field of a protocol data unit, encrypting a file in a manner independent of the "i-node" of the file, or encrypting a disk sector independent of the position of the sector in the physical media. The first example typically arises when there is some fixed communications protocol that has left some number of message bits available, yet provided no security. It is desired to add security but without changing the number of bits of each field. In other words, it is desired to be able to send, encrypted, all of the message which was formally transmitted in the clear.

It should be appreciated by those skilled in the art that the specific embodiments disclosed above may be readily utilized as a basis for modifying or designing other routines for carrying out the same purposes of the present invention.

#### Claims

1. A method, using first and second keys, to encrypt a plaintext string to a ciphertext string, comprising the steps of:

- (a) cipher block chaining (CBC) the plaintext string using the first key and a first initialization vector (IV) to generate a CBC message authentication code;
- (b) cipher block chaining the plaintext string using the second key and the CBC message authentication code as a second initialization vector to generate an enciphered string; and
- (c) combining the CBC message authentication code and a predetermined portion of the enciphered string to form the ciphertext string.

2. A method according to claim 1 wherein said CBC message authentication code has a length equal to a block length.

3. A method according to claim 1 or claim 2, wherein the predetermined portion of the enciphered string includes all but a last block of the enciphered string.

4. A method according to any one of claims 1 to 3, wherein the first initialization vector is the null vector.

5. A method according to any one of the preceding claims, wherein the plaintext string has a length that is a multiple of a length of a block.

6. A method according to any one of claims 1 to 4, wherein the plaintext string has a length that is not equal to a multiple of a length of a block.

7. A method according to any one of the preceding claims, wherein said step (c) concatenates the CBC message authentication code and the predetermined portion of the enciphered string to form the ciphertext string.

8. A method according to any one of the preceding claims, wherein the ciphertext string has a length equal to the plaintext string.

9. A method according to any one of the preceding claims, wherein the first and second keys are derived from an underlying secret key.

10. A method, using first and second keys and a block cipher, to decrypt a ciphertext string into a plaintext string, the ciphertext string comprising a CBC message authentication code and an enciphered string, comprising the steps of:

- (a) decrypting by cipher block chaining the enciphered string using the second key and the CBC message authentication code as an initialization vector to generate a deciphered string;
- (b) cipher block chaining the deciphered string using the first key and a null initialization vector to generate a string having a last block;
- (c) calculating a predetermined function of the last block and an inverse of the block cipher under the first key at the CBC message authentication code; and
- (d) combining the deciphered string and a result of the predetermined function to generate the plaintext string.

11. A method according to claim 10 wherein the block cipher is DES.

12. A method according to claim 10 or claim 11, wherein the predetermined function in step (c) is an exclusive OR.

13. A computer apparatus, comprising:

a storage device;  
 program means supported in the storage device for encrypting a plaintext string into a ciphertext string, the  
 5 program means comprising:  
 means for cipher block chaining (CBC) the plaintext string using the first key and a first initialization vector (IV) to generate a CBC message authentication code;  
 means for cipher block chaining the plaintext string using the second key and the CBC message authentication code as a second initialization vector to generate an enciphered string; and  
 10 means for combining the CBC message authentication code and a predetermined portion of the enciphered string to form the ciphertext string.

14. A computer apparatus, comprising:

15 a storage device;  
 program means supported in the storage device for decrypting a ciphertext string into a plaintext string, the ciphertext string comprising a CBC message authentication code and an enciphered string, the program means comprising:  
 20 means for decrypting the enciphered string by cipher block chaining the enciphered string using a secret key and the CBC message authentication code as an initialization vector to generate a deciphered string;  
 means for cipher block chaining the deciphered string using a second secret key and a null initialization vector to generate a string having a last block;  
 means for calculating a predetermined function of the last block and an inverse of a block cipher evaluated  
 25 using the second secret key; and  
 means for combining the deciphered string and the predetermined function to generate the plaintext string.

15. A program storage device readable by a processor and tangibly embodying a program of instructions executable by the processor to perform encryption and decryption methods, using first and second keys and a block cipher, wherein the encryption method comprises the steps of:

30 (a) cipher block chaining (CBC) a plaintext string using the first key and an initialization vector (IV) to generate a CBC message authentication code;  
 (b) cipher block chaining the plaintext string using the second key and the CBC message authentication code as the initialization vector to generate an enciphered string; and  
 35 (c) combining the CBC message authentication code and a portion of the enciphered string to form a ciphertext string;

and wherein the decryption method comprises the steps of:

40 (a) cipher block chaining the enciphered string using the second key and the CBC message authentication code as the initialization vector to generate a deciphered string;  
 (b) cipher block chaining the deciphered string using the first key and a null initialization vector to generate a string having a last block;  
 45 (c) calculating a predetermined function of the last block and an inverse of the block cipher under the first key at the CBC message authentication code; and  
 (d) combining the deciphered string and the predetermined function to generate the plaintext string.



FIG. 1

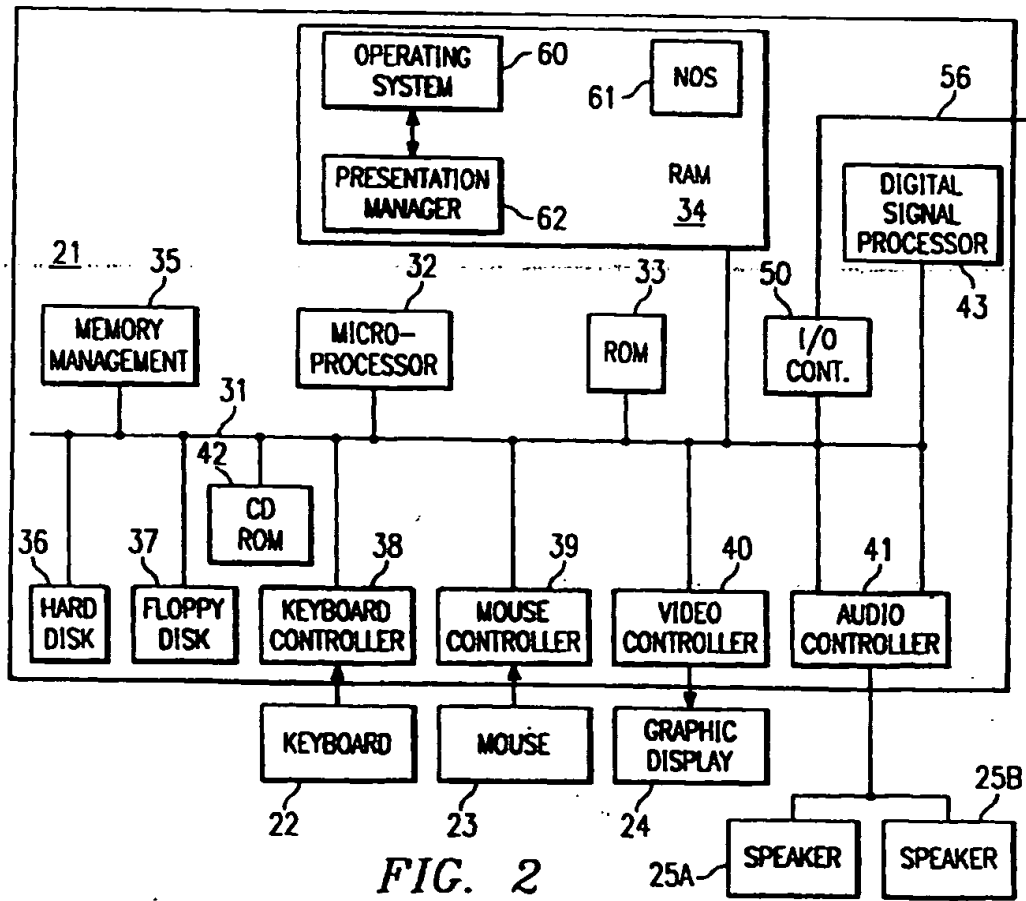
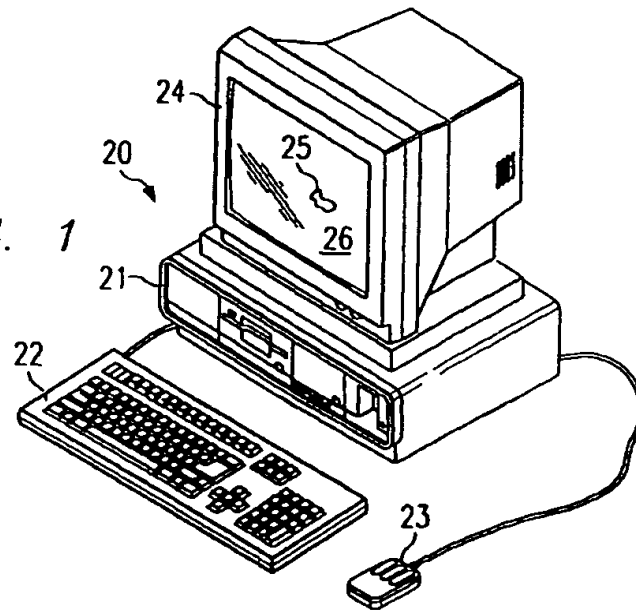
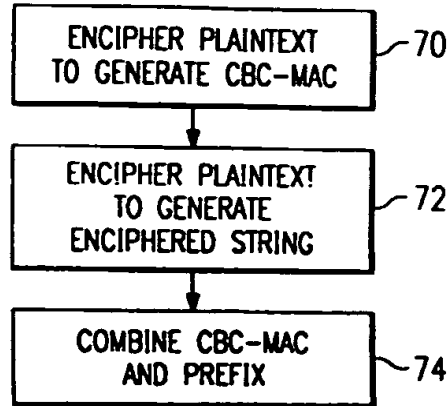


FIG. 2

*FIG. 3*



*FIG. 4*

